Selective Decentralization to Improve Reinforcement Learning in Unknown Linear Noisy Systems

Thanh Nguyen
Dept. of Computer and Information Science
Indiana University - Purdue University - Indianapolis
Indiana, United States
thamnguy@imail.iu.edu

Snehasis Mukhopadhyay
Dept. of Computer and Information Science
Indiana University - Purdue University - Indianapolis
Indiana, United States
smukhopa@cs.iupui.edu

***Abstract*— In this paper, we answer the question of to what extend selective decentralization could enhance the learning and control performance when the system is noisy and unknown. Compared to the previous works in selective decentralization, in this paper, we add the system noise as another complexity in the learning and control problem. Thus, we only perform analysis for some simple toy examples of noisy linear system. In linear system, the Halminton-Jaccobi-Bellman (HJB) equation becomes Riccati equation with closed-form solution. Our previous framework in learning and control unknown system is based on the following principle: approximating the system using identification in order to apply model-based solution. Therefore, this paper would explore the learning and control performance on two aspects: system identification error and system stabilization. Our results show that selective decentralization show better learning performance than the centralization when the noise level is low.**

Keywords—selective decentralization, multi-agent systems, reinforcement learning

I. INTRODUCTION

Decentralized learning, or multi-agent learning, has been one of the emerging topics in intelligent systems in the recent decades [1-3]. Decentralization decouples the entire system's state variables into subsystems using domain knowledge or partition techniques, and assigns an agent for each subsystem. Each agent is responsible to learn on the assigned subsystem. Decentralized learning offers several advantages over centralized learning, such as parallel execution and robustness to agent failure [4]. However, handling the interconnection among the agents is one of the most challenging tasks in decentralized learning in achieving stable learning performance [5, 6].

From the connection handling point of view, there are two popular types of approach in decentralize learning: local connector and central coordinator. In the local connector approach, each learning agent is responsible for its own communication and sharing information [7]. The typical works in this approach use the interconnection parameter to estimate the boundary of the single agent's environment model [7], Q-function [8, 9] or state-utility function [10, 11]. This approach has been known for the time efficiency. However, it requires that the agents must fully or partially know their connections and neighborhoods. In the other hands, the central coordinator approach may not require prior knowledge about the agent neighborhoods. In this approach, all agents send its information to a central coordinator. The central coordinator is responsible to decide how to group the agents into joint actions [12-14]. The central coordinator could be flexible in assigning the joint action many agents. The number of possible joint structures is bounded by the subset selection problem [15], where the solution space grows exponentially according to the Bell number [16].

In the recent years, we have developed and refined the selective decentralization framework to tackle some decentralized reinforcement learning and control problems [17, 18]. The framework tackles problems when the learning agents completely do not know the system, except their assigned components and their learning objectives. Briefly, the key theme in selective decentralization is the joint structures in which the subsets of agents fully cooperate to learn the optimal actions. A central coordinator unit decides which decentralization structure could provide the best learning performance. The metric used by the central coordinator to select the best structure depends on both the nature of the problem and the control technique used in the problem. For example, in [17], the central coordinator chooses system identification error as the metric because the learning and control technique is model-based. In [18], since Q-learning is the learning technique, the central coordinator selects the structure maximizing the cumulative Q-value gained to perform the subsequence learning and control steps. It is expected that selective decentralization could learn the best decentralization structure for learning and control while performing these tasks.

This paper adds the system noise as another dimension of complexity in the learning and control problem for the selective decentralization framework. To simplify the overall problem and focus more on the noise impact, we only perform experiments and evaluations on the linear-quadratic-regulator (LQR) problem. In the LQR problem, the HJB equation – one of the central theme in reinforcement learning – transforms to the Riccati equation with closed-form solution [19]. We want to answer the following questions. First, to what extend the selective decentralization could improve the system identification, compared to the centralized approach, given increasing level of noise? Second, to what extend the selective decentralization could stabilize the system faster than the centralized approach could, given increasing level of noise? Due to the complexity of convergence analysis, we are not able to

This is the author's manuscript of the article published in final edited form as:
Nguyen, T., & Mukhopadhyay, S. (2017). Selective decentralization to improve reinforcement learning in unknown linear noisy systems. In 2017 21st Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES) (pp. 77–82). https://doi.org/10.1109/IESYS.2017.8233565

prove the expected performance gained by the selective decentralization. Therefore, this paper is mostly to confirm the results of our approach in some toy examples.

II. METHOD

A. Problem statement

1) The LQR learning problem

The LQR system in this paper has the form:

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{r}(t) \quad (1)$$

where $\mathbf{x} \in \Re^n$ is the state vector, $\mathbf{u} \in \Re^n$ is the action (also called control) vector, $\mathbf{r} \in \Re^n$ is a random unknown noise vector with expected value of $\mathbf{0}$, $\mathbf{A} \in \Re^{n \times n}$ is the state-transition matrix, which is unknown, and $\mathbf{B} \in \Re^{n \times n}$ is a known semi positive-definite matrix. We assume that $\mathbf{r}$ are under a multivariate normal distribution. Here, we set $\mathbf{x}$ and $\mathbf{u}$ to have the same dimensionality for the ease of decentralization. The main objective is to learn the sequence of action unit $\mathbf{u}$ to stabilize $\mathbf{x}$:

$$\mathbf{x}(t) \to 0, \mathbf{u}(t) \to 0 \text{ as } t \to \infty \quad (2)$$

To apply the control theory as the learning algorithm, we convert the objective in (2) into a more formal control problem: minimizing the feedback function:

$$J(\mathbf{x}(0)) = \sum_{t=0}^{\infty} \left(\mathbf{x}(t)^T \mathbf{Q} \mathbf{x}(t) + \mathbf{u}(t)^T \mathbf{R} \mathbf{u}(t)\right) \quad (3)$$

where $\mathbf{Q}$ and $\mathbf{R}$ are known positive-definite matrices.

Since the Riccati-equation method in solving LQR problem is model-based, we need to find the approximation matrix $\widehat{\mathbf{A}} \in \Re^{n \times n}$ for $\mathbf{A}$ such that with the predicted state vector:

$$\hat{\mathbf{x}}(t+1) = \widehat{\mathbf{A}}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (4)$$

the identification error:

$$e(t) = |\mathbf{x}(t) - \hat{\mathbf{x}}(t)|^2 \quad (5)$$

approaches 0 as t $\to \infty$. In decentralized learning, let k be the number of agents in (1) with dimension n_1, n_2, … n_k such that $\sum_{i=1}^{k} n_i = n$. A decentralized structure computes $\widehat{\mathbf{A}}$ in block-diagonal matrix form:

$$\widehat{\mathbf{A}} = \begin{bmatrix} \widehat{\mathbf{A}_1} & & & \\ & \widehat{\mathbf{A}_2} & & \\ & & \ddots & \\ & & & \widehat{\mathbf{A}_k} \end{bmatrix} \quad (6)$$

B. Two phases in model-based selective decentralization learning

To derive a model-based solution, the selective decentralization technique includes two phases. First, the identification phase provides the environment approximators for all agents. Second, the action phases computes the agents' optimal action regarding the most updated environment approximators. In problem (1), the details for these two phases are as follow.

1) Linear system identification

The theory for linear time-invariant system identification has been well-studied. The gradient descent is one of the most robust methods as shown in [20]. With a random initial approximation matrix $\widehat{\mathbf{A}}(0)$, we have

$$\widehat{\mathbf{A}}(t) = \widehat{\mathbf{A}}(t-1) - \alpha \frac{\left(\mathbf{x}(t) - \hat{\mathbf{x}}(t)\right)\mathbf{x}(t-1)^T}{1 + \mathbf{x}(t-1)^T \mathbf{x}(t-1)} \quad (7)$$

More details on deriving (7) could be found in [17]. Here, the noise element $\mathbf{r}(t)$ does not appear in (7) since (7) is in linear form and the expected value of $\mathbf{r}(t)$ is 0.

2) Action derived from control system solution

As we have known, equations (1-3) forms the Riccati equation [21] with variable $\mathbf{P}$

$$\mathbf{A}^T\mathbf{P}\mathbf{A} - \mathbf{P} - \mathbf{A}^T\mathbf{P}\mathbf{B}(\mathbf{B}^T\mathbf{P}\mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^T\mathbf{P}\mathbf{A} + \mathbf{Q} = 0 \quad (12)$$

The solution $\mathbf{P}$ could be found by DARE algorithm implemented in [22]. At each iteration, by replacing $\mathbf{A}$ by $\widehat{\mathbf{A}}(t)$ in (7), we find $\widehat{\mathbf{P}}(t)$ such that:

$$\widehat{\mathbf{A}}(t)^T\widehat{\mathbf{P}}(t)\widehat{\mathbf{A}}(t) - \widehat{\mathbf{P}}(t) \ldots$$

$$-\widehat{\mathbf{A}}(t)^T\widehat{\mathbf{P}}(t)\mathbf{B}\left(\mathbf{B}^T\widehat{\mathbf{P}}(t)\mathbf{B} + \mathbf{I}\right)^{-1}\mathbf{B}^T\widehat{\mathbf{P}}(t)\widehat{\mathbf{A}}(t) + \mathbf{I} = 0 \quad (9)$$

and the action vector $\mathbf{u}(t)$ is computed by

$$\mathbf{u}(t) = -\left(\mathbf{I} + \mathbf{B}^T\widehat{\mathbf{P}}(t)\mathbf{B}\right)^{-1}\mathbf{B}^T\widehat{\mathbf{P}}(t)\widehat{\mathbf{A}}(t)\mathbf{x}(t) \quad (10)$$

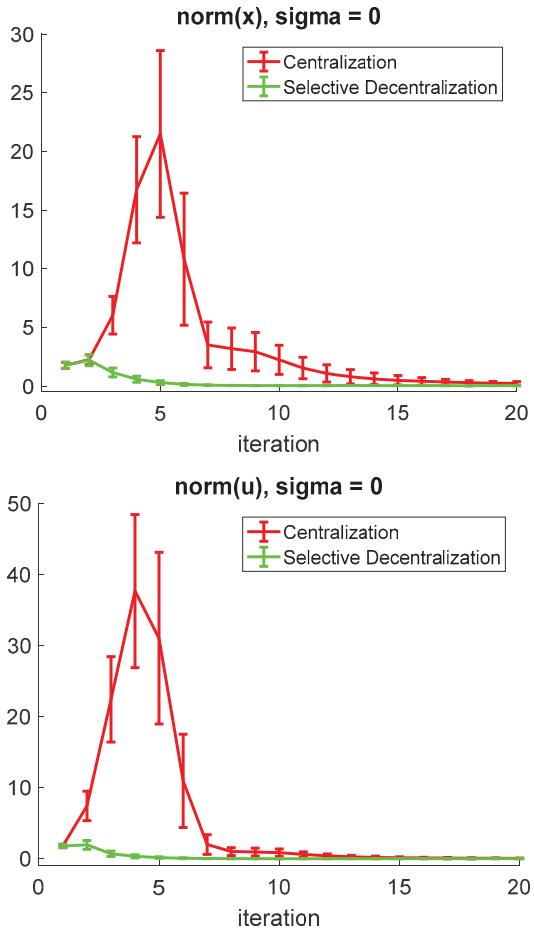


Fig. 1. Comparison of learning performance between the centralized systems and the selectively decentralized systems when the systems is are completely decoupled (σ=0).

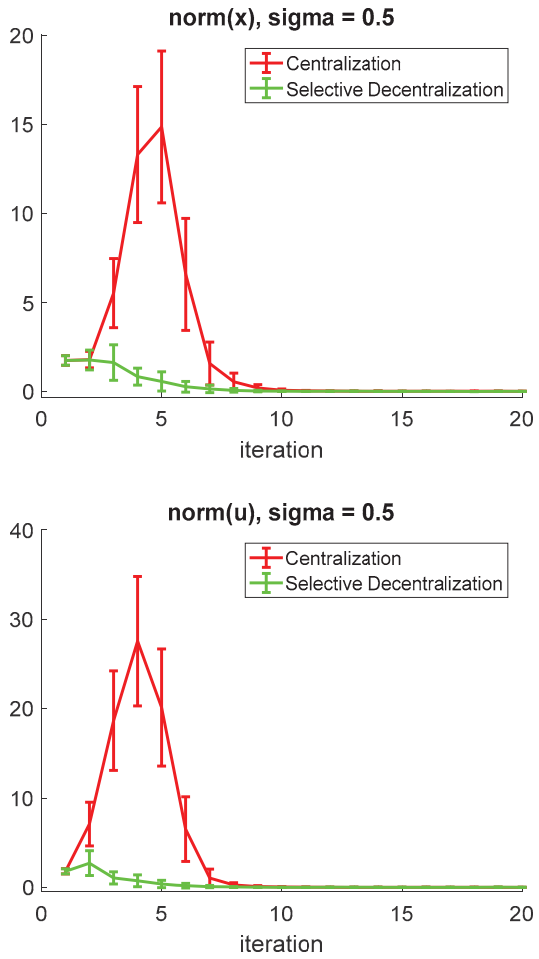


Fig. 2. Comparison of learning performance between the centralized systems and the selectively decentralized systems when the systems are strongly coupled (σ=0.5).

C. *Selective decentralization in LQR problem (1)*

As we mentioned, for the system of k components, the number of possible decentralization structures is the k^{th} Bell's number [16]. Since the details and examples for selective decentralization could be found in [17, 18], this section only provides a brief example to demonstrate the selective decentralization paradigm and the parameters needed during applying the technique. For example, with $k = 3$, we have $B(k) = 5$ possible decentralization structures: {{1,2,3}}, {{1,2}, {3}}, {{1,3}, {2}}, {{1}, {2,3}} and {{1}, {2}, {3}}. It is easy to see that structure {1,2,3} corresponds to centralization and structure {{1}, {2}, {3}} corresponds to completely decentralization. With scheme {{1, 2}, {3}}, we compute $\widehat{\mathbf{A}}$ as $\begin{bmatrix} \widehat{\mathbf{A}}_{1,2} & \\ & \widehat{\mathbf{A}}_3 \end{bmatrix}$. Here, $\widehat{\mathbf{A}}_{1,2}$ is computed only using $\mathbf{x}_1$,$\mathbf{x}_2$, $\mathbf{u}_1$ and $\mathbf{u}_2$ according to formula (9-10); meanwhile $\widehat{\mathbf{A}}_3$ is computed only using $\mathbf{x}_3$ and $\mathbf{u}_3$.

Similar to [17] we use the ***window-based identification error*** to select the best decentralization scheme. Let Ω be the time-window size and w be the window index. Window w include the time from $t = (w\text{-}1)\,\Omega + 1$ to $t = w\Omega$. Let $E(w)$ be the window-based identification error at window w, which is the average of $e(t)$ within window w. The pseudo code for selective decentralization is as follow

initialize b^*: the best decentralization structure
  $\widehat{\mathbf{A}}(0)$: random matrix.
    Each structure could initiate its own $\widehat{\mathbf{A}}(0)$
for t from 1 to the maximum time index
  receive $\mathbf{x}(t)$
  calculate control $\mathbf{u}(t)$ **using** b^* and (9-10)
  for each decentralization structure b
    identify $\widehat{\mathbf{A}}(t)$ according to (7)
    store identification error computed by (4-5)
  end for
  if t % $\Omega = 0$ // end of a window
    //select the new best decentralization structure
    Select the decentralization structure with the lowest $E(w)$ as b^*
    Clear the records of previous identification error
  end if
end for

The key point in selective decentralization is choosing the decentralization scheme b for later computation. For the example of $k = 5$ above, we must ensure that the identification error of scheme {{1, 2}, {3}} is the lowest so that we can compute the identification as $\widehat{\mathbf{A}} = [\widehat{\mathbf{A}}_{1,2}, \widehat{\mathbf{A}}_3]$, and control $\mathbf{u} = [\mathbf{u}_{1,2}, \mathbf{u}_3]$

In addition, we call the decentralization where the subsystems have no communication as *complete decentralization*. In this scheme, the agents do not cooperate with the other. Obviously, the selective decentralization technique examines complete decentralization. Also, selective decentralization also examines *centralization*, when all agents join together in identification and action phases.

III. Simulation Case Studies

In this paper, we setup systems of 10 dimensions with $k = 5$ for equation (1). The matrix $\mathbf{A}$ is setup with underlying subsystem components {{1,2}, {3, 4},{5, 6}, {7,8}, {9, 10}} as follow:

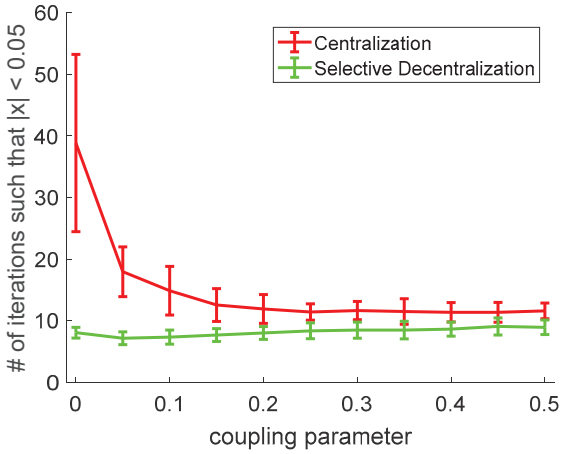
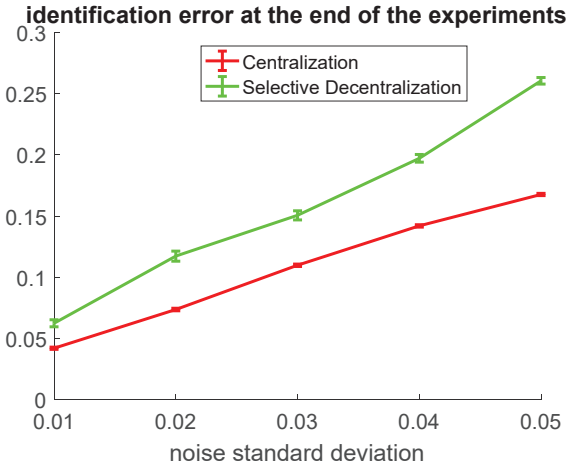


Fig. 3. Number of iterations needed to bring norm($\mathbf{x}$) < 0.05

$$\mathbf{A}=\begin{bmatrix} 0.68 & 0.32 & & & & & & & & \\ 0.20 & 0.80 & & & & & & & & \\ & & 0.25 & 0.75 & & & & \sigma & & \\ & & 0.44 & 0.56 & & & & & & \\ & & & & 0.50 & 0.50 & & & & \\ & & & & 0.41 & 0.59 & & & & \\ & & & & & & 0.85 & 0.15 & & \\ & & \sigma & & & & 0.15 & 0.85 & & \\ & & & & & & & & 0.35 & 0.65 \\ & & & & & & & & 0.67 & 0.33 \end{bmatrix} \quad (11)$$

where the non-block entries of **A** are a random numbers between 0 and σ, which is called coupling parameters. The initial control variables **u**(0) and state variable **x**(0) are set randomly between -1 and 1. As shown in (10), σ represents the underlying interconnection among the subsystems. We experiment with σ from 0 to 0.5; in the other word, from the completely decoupled system to the strongly coupled system. To avoid numerical overflowing, we normalize **A** into a Markov matrix in (1). We set **B** as the identity matrix. For identification, we set $\alpha = 1$. At the starting point, we set all elements of **x**(0) as random numbers between -1 and 1. Each noise element is randomly generated from normal distribution with mean of 0 and small standard deviation of from 0.01 to 0.05. Higher standard deviation implies more noise. For statistical purposes, we repeat the experiment 50 times for each choice of coupling parameter. We set the window size $w = 1$. For time index t, we terminate the experiment at $t = 500$. This choice is made based on the observation in [17], when the system is stabilized after the first tens iterations.



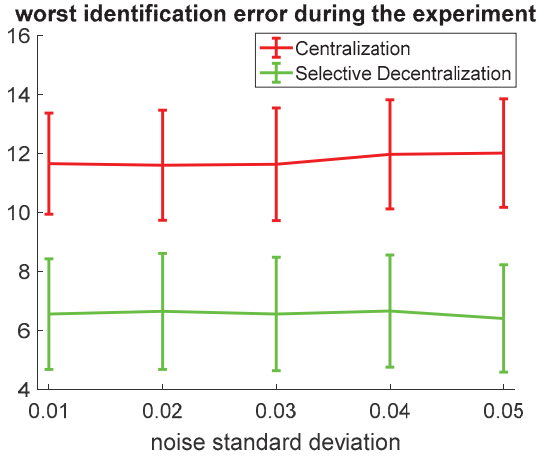
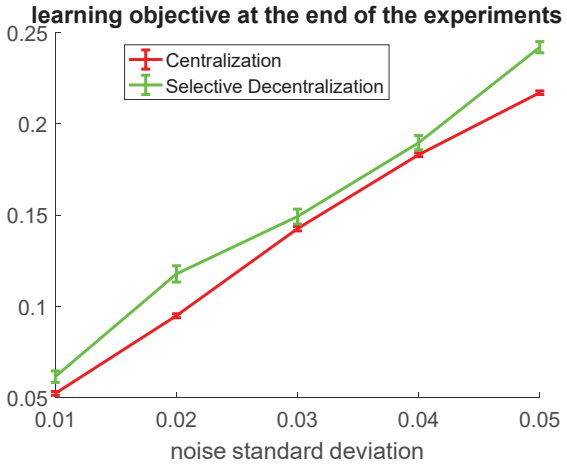


Fig. 4. Comparison of identification errors between the selectively decentralized approach and the centralized approach given increasing noise level



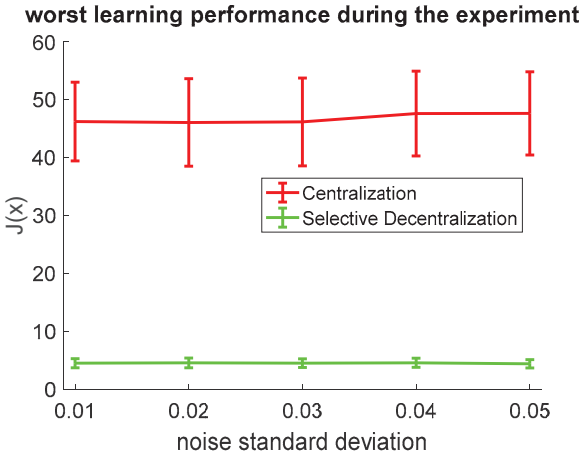


Fig. 5. Comparison of learning performance $J(\mathbf{x})$ between the selectively decentralized approach and the centralized approach given increasing noise level

Figures 1 and 2 show that the selectively decentralized system shows better control performance than the centralized system. In these figures, we draw the result when the noise standard deviation is 0.01. Figure 1 shows the result when (1) is completely decoupled (σ=0). Figure 2 shows the result when (1) is the most coupled in our experiments (σ=0.5). We use norm(**x**) and norm(**u**) to denote the second-norm of **x** and **u**. In these figures, the numbers for norm(**x**) and norm(**u**) are the average values of the 50 random repetitions.

Notably, the centralization and selective decentralization can learn how to stabilize (1) despite the noise. However, we see significant gap between the performance of the centralization and the performance of selective decentralization. Selective decentralization stabilizes (1) faster. This gap tends to decrease when the systems are more coupled. Figure 3 shows the average

number of iteration for the centralization selective decentralization to bring norm(**x**) less than 0.05. We observe that the selective decentralization always outperforms the other approaches regardless of the coupling parameters.

In figures 4 and 5, we show the learning performance of the selectively decentralized approach and the centralized approach when the noise increases. As mentioned, since each noise element is a random number with mean 0, higher noise standard deviation implies more noise. In figure 4, we measure the average identification error (5) at the last 50 iterations during the experiments as the 'identification error at the end of the experiment', and the largest identification error from $t = 1$ to $t = 500$ as the 'worst identification error'. Similar to figure 4, in figure 5, we measure the learning objective (3). Overall, after the experiment, we observe that the centralized approach is robust against increasing level of noise. Here, the convergent identification error and learning objective in the centralized approach do not degrade when more noise is introduced. Meanwhile, these metrics in the selectively decentralized approach deteriorate linearly with the increasing noise. However, the selectively decentralized approach still significantly outperforms the centralized approach when we look at the worst identification error and learning objective.

IV. CONCLUSIONS

In this paper, we show that selective decentralization can improve the learning performance in both linear and nonlinear systems with several levels of interconnection among subsystems when a low level of noise exists. Here, we measure the performance on the number of iterations, or samples, needed in learning. This measurement of performance is useful for problems in which the number of training samples is limited. However, this improvement is a trade-off between the 'best end point' and the 'cost to get the end point'. As figures 3-5 suggest, the selective decentralized approach, which drops some of the interconnection information for faster convergence and lower absolute-instant cost $J(\mathbf{x})$, may terminate with poorer learning objective, compared to the centralized approach.

We also observe that the learning performance of the selective decentralization approach deteriorates when more noise is introduced into the system. Figures 4 and 5 may suggest that the deterioration is caused by the poorer identification error. Therefore, noise-filtering techniques should be applied in combination with the selective decentralization approach to reduce system identification error.

There are several limitation in this paper. First, we are not able to offer an analysis of result to explain the performance gap between the selective decentralization and the centralization approaches. Therefore, to avoid the possible bias toward example-specific, we repeated the experiment with random parameters in 50 times. Second, the case study is limited in linear problem, where solutions are theoretically proven. Third, in selective decentralization, we still explore all possible decoupling scheme $B(k)$, which grows exponentially. However, when in the data scarcity scenario, such as biological systems where the funding is usually available for no more than 30 experiments, we may have to exchange computational cost for the short convergence, In this case, we strongly believe that the selective decentralization is helpful.

ACKNOWLEDGMENT

The research presented in this paper was supported by a National Science Foundation grant (No. ECCS-1407925)

REFERENCES

[1] Mesbah, M., and Su, R.: 'Decentralized learning control', in 'Book Decentralized learning control' (1992, edn.), pp. 1327-1332 vol.1322

[2] Panait, L., and Luke, S.: 'Cooperative multi-agent learning: The state of the art', Autonomous agents and multi-agent systems, 2005, 11, (3), pp. 387-434

[3] Sastry, P., Phansalkar, V., and Thathachar, M.: 'Decentralized learning of Nash equilibria in multi-person stochastic games with incomplete information', IEEE Transactions on systems, man, and cybernetics, 1994, 24, (5), pp. 769-777

[4] Busoniu, L., De Schutter, B., and Babuska, R.: 'Decentralized reinforcement learning control of a robotic manipulator', Proc. Control, Automation, Robotics and Vision, 2006. ICARCV'06. 9th International Conference on, pp. 1-6

[5] Ioannou, P.A.: 'Decentralized adaptive control of interconnected systems', Automatic Control, IEEE Transactions on, 1986, 31, (4), pp. 291-298

[6] Gavel, D.T., and Siljak, D.: 'Decentralized adaptive control: structural conditions for stability', Automatic Control, IEEE Transactions on, 1989, 34, (4), pp. 413-426

[7] Viseras, A., Wiedemann, T., Manss, C., Magel, L., Mueller, J., Shutin, D., and Merino, L.: 'Decentralized multi-agent exploration with online-learning of Gaussian processes', in: Proc. Robotics and Automation (ICRA), 2016 IEEE International Conference on, pp. 4222-4229

[8] Chen, X., Fu, B., He, Y., and Wu, M.: 'Timesharing-tracking framework for decentralized reinforcement learning in fully cooperative multi-agent system', IEEE/CAA Journal of Automatica Sinica, 2014, 1, (2), pp. 127-133

[9] Chu, T., Qu, S., and Wang, J.: 'Large-scale traffic grid signal control with regional Reinforcement Learning', in: Proc. American Control Conference (ACC), 2016, pp. 815-820

[10] Liu, D., Wang, D., and Li, H.: 'Decentralized stabilization for a class of continuous-time nonlinear interconnected systems using online learning optimal control approach', IEEE transactions on neural networks and learning systems, 2014, 25, (2), pp. 418-428

[11] Liu, D., Wang, D., and Li, H.: 'Online learning optimal control for decentralized stabilization of nonlinear interconnected systems', in: Proc. Cyber Technology in Automation, Control and Intelligent Systems (CYBER), 2013 IEEE 3rd Annual International Conference on, pp. 229-234

[12] Ruan, X., Bien, Z.Z., and Park, K.-H.: 'Decentralized iterative learning control to large-scale industrial processes for nonrepetitive trajectory tracking', IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2008, 38, (1), pp. 238-252

[13] Cherukuri, A., and Cortýs, J.: 'Decentralized Nash equilibrium learning by strategic generators for economic dispatch', in: Proc. American Control Conference (ACC), 2016, pp. 1082-1087

[14] Li, J., and Zhangt, J.: 'Using estimation of distribution algorithm to coordinate decentralized learning automata for meta-task scheduling', in: Proc. Evolutionary Computation (CEC), 2014 IEEE Congress on, pp. 2077-2084

[15] Tilak, O., and Mukhopadhyay, S.: 'Decentralized and partially decentralized reinforcement learning for distributed combinatorial optimization problems', in: Proc. Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on, pp. 389-394

[16] Rota, G.-C.: 'The number of partitions of a set', The American Mathematical Monthly, 1964, 71, (5), pp. 498-504

[17] Nguyen, T., and Mukhopadhyay, S.: 'Identification and Optimal Control of Large-scale System Using Selective Decentralization'. Proc. IEEE International Conference on Systems. Man and Cybernetics, Budapest2016

[18] Nguyen, T., and Mukhopadhyay, S.: 'Selectively Decentralized Q-Learning'. Proc. IEEE International Conference on Systems, Man, and Cybernetics, Bannf, Canada2017

[19] Bellon, J.: 'Riccati Equations in Optimal Control Theory', 2008

[20] Keesman, K.J.: 'System Identification: an Introduction' (Springer-Verlag, 2011. 2011)

[21] Lancaster, P., and Rodman, L.: 'Algebraic riccati equations' (Clarendon press, 1995. 1995)

[22] Arnold III, W.F., and Laub, A.J.: 'Generalized eigenproblem algorithms and software for algebraic Riccati equations', Proceedings of the IEEE, 1984, 72, (12), pp. 1746-1754